

Computer Lab 04: RandomLocationMoverManager, Constant Rate Sensor

RandomLocationMoverManager

A Mover Manager's responsibility is to manage the movement behavior for a single Mover instance. Each type of Mover Manager has its own rules for this behavior. You have seen the PathMoverManager that sends a Mover along a list of waypoints. The RandomLocationMoverManager's movement algorithm chooses a random point according to two RandomVariate instances, and sends its Mover to that location. When the Mover arrives, another random destination is chosen, and the process continues until the RandomLocationMoverManager is told to stop (or the simulation ends).

The RandomLocationMoverManager class is a SimEntityBase so it can be a SimEventListener to its Mover. Thus, it "knows" when the Mover has reached its destination when it hears the EndMove(Mover) event. It has the following parameters (private instance variables with setters and getters)::

A Mover - the Mover instance it is managing.

A RandomVariate[] array of length 2 (called nextLocation) that is used to generate the random locations.¹

A boolean to indicate whether the RandomLocationMoverManager should start its cycle on the Run event (startOnReset).

Also, there is a single state variable (protected with a getter but no setter), a boolean to indicate whether it is running or not (running).

The constructor should have signature (Mover, RandomVariate[]), and setStartOnReset should be false by default. There should be setters and getters for all four instance variables. The setter for the Mover should establish the listening for the new mover and remove the listening from the old mover (if any). Thus, it should look like this:²

```
public void setMover(Mover newMover) {
    if (mover != null) {
        mover.removeSimEventListener(this);
    }
    mover = newMover;
    mover.addSimEventListener(this);
}
```

For convenience, define a protected method getNextLocation() that returns a random Point2D instance.³ This return should look like this:

```
return new Point2D.Double(nextLocation[0].generate(), nextLocation[1].generate());
```

where nextLocation is the RandomVariate[] array.

The start() method sets running to true and tells the Mover to moveTo a random location. The stop() method sets running to false and invokes stop() on the Mover. The doRun() method invokes start() if startOnReset is true. Finally, doEndMove() checks to see whether the Mover passed in is indeed the Mover belonging to the RandomLocationMoverManager instance. If so, then it simply tells the Mover to move to another randomly generated location.

-
1. The setter and getter should clone the array.
 2. Also see the code for PathMoverManager
 3. You will need to import java.awt.geom.*;

After writing your `RandomLocationMoverManager` class, test it using random locations uniformly distributed in a square 200×200 . Create a `RandomVariate[]` array in `main()` with each element `Uniform(-100, 100)`. Seed the first `RandomVariate` with `CongruentialSeeds.SEED[0]` and the second with `CongruentialSeeds.SEED[1]`. Instantiate two `UniformLinearMovers` at `(-100, -100)` and `(100, 100)` with speeds `30.0` and `40.0`, respectively. Instantiate a `RandomLocationMoverManager` for each one, using the same `RandomVariate[]` array for each Mover Manager. Set `startOnReset` to `true` for each Mover Manager and run the simulation in verbose mode for `15.0` time units. Your output should be as follows.

Output 1

```

** Event List -- Starting Simulation **
0.000    Run
0.000    Run
15.000   Stop
** End  of Event List -- Starting Simulation **

Time: 0.000    Current Event: Run      [1]
** Event List -- **
0.000    Run
0.000    StartMove      {Fred (-100.000,-100.000) [29.419,5.876]}
15.000   Stop
** End  of Event List -- **

Time: 0.000    Current Event: Run      [2]
** Event List -- **
0.000    StartMove      {Fred (-100.000,-100.000) [29.419,5.876]}
0.000    StartMove      {Barney (100.000,100.000) [-28.405,-28.163]}
15.000   Stop
** End  of Event List -- **

Time: 0.000    Current Event: StartMove {Fred (-100.000,-100.000) [29.419,5.876]}
[1]
** Event List -- **
0.000    StartMove      {Barney (100.000,100.000) [-28.405,-28.163]}
6.285    EndMove        {Fred (-100.000,-100.000) [29.419,5.876]}
15.000   Stop
** End  of Event List -- **

Time: 0.000    Current Event: StartMove {Barney (100.000,100.000) [-28.405,-28.163]}
[2]
** Event List -- **
5.786    EndMove        {Barney (100.000,100.000) [-28.405,-28.163]}
6.285    EndMove        {Fred (-100.000,-100.000) [29.419,5.876]}
15.000   Stop
** End  of Event List -- **

Time: 5.786    Current Event: EndMove   {Barney (-64.341,-62.936) [3.722,39.826]}      [1]
** Event List -- **
5.786    StartMove      {Barney (-64.341,-62.936) [3.722,39.826]}
6.285    EndMove        {Fred (70.205,-66.006) [29.419,5.876]}
15.000   Stop
** End  of Event List -- **

Time: 5.786    Current Event: StartMove {Barney (-64.341,-62.936) [3.722,39.826]}
[3]
** Event List -- **
6.285    EndMove        {Fred (70.205,-66.006) [29.419,5.876]}
9.249    EndMove        {Barney (-64.341,-62.936) [3.722,39.826]}
15.000   Stop

```

```

** End of Event List -- **

Time: 6.285      Current Event: EndMove   {Fred (84.906,-63.069) [12.750,-27.156]}      [2]
** Event List -- **
6.285   StartMove      {Fred (84.906,-63.069) [12.750,-27.156]}
9.249   EndMove        {Barney (-62.481,-43.034) [3.722,39.826]}
15.000  Stop
** End of Event List -- **

Time: 6.285      Current Event: StartMove      {Fred (84.906,-63.069) [12.750,-27.156]}
[4]
** Event List -- **
7.098   EndMove        {Fred (84.906,-63.069) [12.750,-27.156]}
9.249   EndMove        {Barney (-62.481,-43.034) [3.722,39.826]}
15.000  Stop
** End of Event List -- **

Time: 7.098      Current Event: EndMove   {Fred (95.274,-85.151) [-28.749,8.573]} [3]
** Event List -- **
7.098   StartMove      {Fred (95.274,-85.151) [-28.749,8.573]}
9.249   EndMove        {Barney (-59.455,-10.649) [3.722,39.826]}
15.000  Stop
** End of Event List -- **

Time: 7.098      Current Event: StartMove      {Fred (95.274,-85.151) [-28.749,8.573]} [5]
** Event List -- **
9.249   EndMove        {Barney (-59.455,-10.649) [3.722,39.826]}
12.176  EndMove        {Fred (95.274,-85.151) [-28.749,8.573]}
15.000  Stop
** End of Event List -- **

Time: 9.249      Current Event: EndMove   {Barney (-51.450,75.011) [32.474,-23.354]}      [4]
** Event List -- **
9.249   StartMove      {Barney (-51.450,75.011) [32.474,-23.354]}
12.176  EndMove        {Fred (33.440,-66.713) [-28.749,8.573]}
15.000  Stop
** End of Event List -- **

Time: 9.249      Current Event: StartMove      {Barney (-51.450,75.011) [32.474,-23.354]}
[6]
** Event List -- **
12.176  EndMove        {Fred (33.440,-66.713) [-28.749,8.573]}
12.999  EndMove        {Barney (-51.450,75.011) [32.474,-23.354]}
15.000  Stop
** End of Event List -- **

Time: 12.176     Current Event: EndMove   {Fred (-50.693,-41.626) [18.616,23.525]}      [5]
** Event List -- **
12.176  StartMove      {Fred (-50.693,-41.626) [18.616,23.525]}
12.999  EndMove        {Barney (43.584,6.666) [32.474,-23.354]}
15.000  Stop
** End of Event List -- **

Time: 12.176     Current Event: StartMove      {Fred (-50.693,-41.626) [18.616,23.525]}
[7]
** Event List -- **
12.999  EndMove        {Barney (43.584,6.666) [32.474,-23.354]}
15.000  Stop
16.596  EndMove        {Fred (-50.693,-41.626) [18.616,23.525]}
** End of Event List -- **

Time: 12.999     Current Event: EndMove   {Barney (70.321,-12.562) [-1.515,39.971]}      [6]

```

```

** Event List -- **
12.999 StartMove      {Barney (70.321,-12.562) [-1.515,39.971]}
15.000 Stop
16.596 EndMove        {Fred (-35.366,-22.257) [18.616,23.525]}
** End of Event List -- **

Time: 12.999 Current Event: StartMove      {Barney (70.321,-12.562) [-1.515,39.971]}
[8]
** Event List -- **
15.000 Stop
15.415 EndMove        {Barney (70.321,-12.562) [-1.515,39.971]}
16.596 EndMove        {Fred (-35.366,-22.257) [18.616,23.525]}
** End of Event List -- **

Time: 15.000 Current Event: Stop          [1]
** Event List -- **
    << empty >>
** End of Event List -- **

```

ConstantRateSensor and ConstantRateMediator

To implement the constant rate sensor, write a `ConstantRateSensor` class that subclasses `CookieCutterSensor` and adds a parameter `meanTimeToDetection` (similar to the `FooSensor`).

Next, write the `ConstantRateMediator` class. Declare a static (class) variable of type `RandomVariate` and get an instance of an `Exponential (1.0)` in the static constructor¹. Write a static setter for the seed as follows:

```

public static void setSeed(long seed) {
    timeToDetect.getRandomNumber().setSeed(seed);
}

```

Write the `doEnterRange()` and `doExitRange()` methods in manners similar to the `FooMediator`. The time to detection should be the mean time to detection from the `ConstantRateSensor` times an `Exponential (1.0)` random variate. Let the time from `ExitRange` to `Undetect` be 0.0. The time to detect is given by an `Exponential (μ)` random variable. Use the fact that if $X \sim \text{Exponential}(1.0)$ then $\mu X \sim \text{Exponential}(\mu)$.

The test will involve both a `ConstantRateSensor` and a `CookieCutterSensor`. On the Mover at (-100, -100) put a `ConstantRateSensor` with maximum range 30.0 and mean time to detection of 0.5. On the other Mover put a `CookieCutterSensor` with maximum range 50.0. Be sure to register targets and sensors with an instance of `SensorTargetReferee`. Also, be sure to add both mediator types to the `SensorTargetMediatorFactory`. Run for 30.0 time units. Use `CongruentialSeeds.SEED[2]` for the `ConstantRateMediator`. A portion of the output is shown below.

Output 2

After moving around for awhile, the first detection event by the `ConstantRateSensor` on Fred will detect Barney at time 16.468, as follows

```

simkit.smdx.SensorTargetReferee
Sensors:
CookieCutterSensor (50.0) [Barney (100.000,100.000) [0.000,0.000]]

```

1. That is write a method that looks like this:

```

static {
    timeToDetect = RandomVariateFactory.getInstance("Exponential",
        new Object[] { new Double(1.0) } );
}

```

```

ConstantRateSensor (30.0) [Fred (-100.000,-100.000) [0.000,0.000]] - 0.5
Targets:
Fred (-100.000,-100.000) [0.000,0.000]
Barney (100.000,100.000) [0.000,0.000]

...

Time: 16.071    Current Event: EnterRange    {ConstantRateSensor (30.0) [Fred
(21.831,50.024) [18.616,23.525]] - 0.5, Barney (45.744,68.139) [-31.864,-24.180]]}    [1]
** Event List -- **
16.468 Detection    {Contact: [45.744, 68.139]}
16.596 EndMove      {Fred (21.831,50.024) [18.616,23.525]}
16.930 ExitRange    {ConstantRateSensor (30.0) [Fred (21.831,50.024) [18.616,23.525]] -
0.5, Barney (45.744,68.139) [-31.864,-24.180]}
17.219 ExitRange    {CookieCutterSensor (50.0) [Barney (45.744,68.139) [-31.864,-
24.180]], Fred (21.831,50.024) [18.616,23.525]}
18.120 EndMove      {Barney (45.744,68.139) [-31.864,-24.180]}
30.000 Stop
** End of Event List -- **

Time: 16.468    Current Event: Detection      {Contact: [33.122, 58.561]}    [2]
** Event List -- **
16.596 EndMove      {Fred (29.205,59.343) [18.616,23.525]}
16.930 ExitRange    {ConstantRateSensor (30.0) [Fred (29.205,59.343) [18.616,23.525]] -
0.5, Barney (33.122,58.561) [-31.864,-24.180]}
17.219 ExitRange    {CookieCutterSensor (50.0) [Barney (33.122,58.561) [-31.864,-
24.180]], Fred (29.205,59.343) [18.616,23.525]}
18.120 EndMove      {Barney (33.122,58.561) [-31.864,-24.180]}
30.000 Stop
** End of Event List -- **

Time: 16.596    Current Event: EndMove    {Fred (31.591,62.358) [-23.656,-18.450]}    [8]
** Event List -- **
16.596 EndMove      {ConstantRateSensor (30.0) [Fred (31.591,62.358) [-23.656,-18.450]]
- 0.5}
16.596 StartMove    {Fred (31.591,62.358) [-23.656,-18.450]}
16.930 ExitRange    {ConstantRateSensor (30.0) [Fred (31.591,62.358) [-23.656,-18.450]]
- 0.5, Barney (29.037,55.461) [-31.864,-24.180]}
17.219 ExitRange    {CookieCutterSensor (50.0) [Barney (29.037,55.461) [-31.864,-
24.180]], Fred (31.591,62.358) [-23.656,-18.450]}
18.120 EndMove      {Barney (29.037,55.461) [-31.864,-24.180]}
30.000 Stop
** End of Event List -- **

Time: 16.596    Current Event: EndMove    {ConstantRateSensor (30.0) [Fred (31.591,62.358) [-
23.656,-18.450]] - 0.5}    [4]
** Event List -- **
16.596 StartMove    {Fred (31.591,62.358) [-23.656,-18.450]}
16.930 ExitRange    {ConstantRateSensor (30.0) [Fred (31.591,62.358) [-23.656,-18.450]]
- 0.5, Barney (29.037,55.461) [-31.864,-24.180]}
17.219 ExitRange    {CookieCutterSensor (50.0) [Barney (29.037,55.461) [-31.864,-
24.180]], Fred (31.591,62.358) [-23.656,-18.450]}
18.120 EndMove      {Barney (29.037,55.461) [-31.864,-24.180]}
30.000 Stop
** End of Event List -- **
...

```

Deliverables

Your source code plus verbose output for the final version - just include enough of the output to show the EnterRange-Detection-ExitRange-Undetection sequence for the ConstantRateSensor.